**Exercises II**

3.- Write a function that asks the user for data between 0 and 20, and check if that data exists in the matrix, if so write an *. The function will end when the whole matrix has *.

$$A=\begin{pmatrix} 6 & 14 & 13 \\ 3 & 9 & 5 \\ 7 & 1 & 20 \end{pmatrix}$$

**Solution:**

```
exercise3<-function()
{
 A=matrix(c(6,14,13,3,9,5,7,1,20),3,3, byrow=T)
 max=ncol(A)*nrow(A)
 cont=0
 while(cont<max)
 {
   print("Insert a number (0:20)")
   num=scan(,what=numeric(),1)
   print(num)
   while ((num <0 ) | (num>20))
   {
     print("Out of range. Please, Insert a number (0:20)")
     num=scan(,,what=numeric(),1)
   }
   for (i in 1:nrow(A))
    for(j in 1:ncol(A))
      if (A[i,j] ==num)
      {
           A[i,j] = "*"
         cont=cont+1
         print(cont)
         print(max)
      }
  print(A)
 }
}
```

4.- Write a function that asks the user for data between 0 and 20, and check if that data exists in the matrix, if so write an *. The user will be allowed 10 attempts. At the end, the number of correct guesses will be verified and shown on the screen.

$$A=\begin{pmatrix} 6 & 14 & 13 \\ 3 & 9 & 5 \\ 7 & 1 & 20 \end{pmatrix}$$

**Solution:**

```
exercise4<-function()
{
  A=matrix(c(6,14,13,3,9,5,7,1,20),3,3)
  print("You have 10 attempts")
  numCorrect=0
  attempts=0
  while (attempts <10)
  {
    print("Insert a number (0:20)")
    num=scan(,,1)
    while((num <0 ) | (num>20))
    {
      print("Out of Range")
      num=scan(,,1)
    }
    attempts= attempts +1
    for (i in 1:nrow(A))
      for(j in 1:ncol(A))
        if (A[i,j] ==num)
        {
          A[i,j] = "*"
          numCorrect = numCorrect +1
        }
    print(A)
    print(attempts)
  }
  print(A)
  print("Number of correct guesses: ")
  print(numCorrect)
}
```

5.- Write a function that asks the user for data between 0 and 20, and check if that data exists in the matrix, if so write an *. The number of attempts will be passed to the function by parameter. At the end, matrix and the number of correct guesses will be verified and displayed on the screen; In addition, this information will be stored (matrix and number of correct guesses) in a file called "new.txt"

$$A=\begin{pmatrix} 6 & 14 & 13 \\ 3 & 9 & 5 \\ 7 & 1 & 20 \end{pmatrix}$$

**Solution:**

```
exercise5<-function(times)
{
 A=matrix(c(6,14,13,3,9,5,7,1,20),3,3)
 cat("You have ", times, " attempts\n")
 numCorrect=0
 attempts=0
 while (attempts < times)
 {
  print("Insert a number (0:20)")
  num=scan(„1)
  while((num <0 ) | (num>20))
  {
   print("Out of Range")
   num=scan(„1)
  }
  attempts= attempts +1
  for (i in 1:nrow(A))
   for(j in 1:ncol(A))
    if (A[i,j] ==num)
    {
     A[i,j] = "*"
     numCorrect = numCorrect +1
    }
  print(A)
  print(attempts)
 }
 print(A)

 write.table(A, "C://MandT//nuevo.txt")
 dataFile=c("Number of correct guesses: ",  numCorrect)
 write(dataFile,"C://MandT//nuevo.txt",append=TRUE)
}


>exercise5(4)
```

6.- Write a function that asks the user for data between 0 and 20, and check if that data exists in the matrix, if so write an *. The function will end when the whole matrix has *. The function should control that you do not repeat numbers that you have already tried. Also at the end it will show all the numbers you have tried.

$$A=\begin{pmatrix} 6 & 14 & 13 \\ 3 & 9 & 5 \\ 7 & 1 & 20 \end{pmatrix}$$

**Solution:**

```
Ex6<-function()
{
  A<-matrix(c(6,14,13,3,9,5,7,1,20),3,3)
  print(A)
  whole=FALSE
  numwritten=0
  while(whole==FALSE)
  {
    print("insert an element (0- 20)")
    num=scan(,,1)

   # check the number
    while ((num<0)||(num>20))
    {
       print("Number out of range")
       #add to vector because the numbers that are out of range are also
       numwritten =c(numwritten,num)
      num=scan(,,1)
    }
   #check that the number is not repeated in the vector
    cont=1
    norepeat=TRUE
    while((cont<=length(numwritten))&&(norepeat==TRUE))
    {

      if(num== numwritten [cont])
      {
            norepeat=FALSE
            print("You have tried this number")
      }
      else
      {
        norepeat=TRUE
      }
      cont=cont+1
    }

    # Check if the number is in the matrix and if it has * the matrix
    if(norepeat==TRUE)
    {
        # Add to vector
         numwritten =c(numwritten,num)

        # check if the number is in the matrix
        for(i in 1:nrow(A))
                for(j in 1:ncol(A))
                  if(A[i,j]==num)
                  {
                     A[i,j]="*"
```

```
            }

          # check if whole matrix has *
          i=1
          can=TRUE
         while((i<=nrow(A))&&(can==TRUE))
         {
              j=1
              while((j<=ncol(A))&&(can==TRUE))
             {
                if(A[i,j]=="*")
                    j=j+1
               else
                   can=FALSE
               }
                i=i+1
          }

          # check the variable can
          if(can==TRUE)
          {
              whole=TRUE
          }
          print(A)
          print(numwritten)
     }
 }

  print(A)
  print(numwritten)
}
```